

```
SSSSSSSSSSSSSS 000000000 000 RRRRRRRRRRRR RRR TTTTTTTTTTTTTTT 333333333 222222222
SSSSSSSSSSSSSS 000000000 000 RRRRRRRRRRRR RRR TTTTTTTTTTTTTTT 333333333 222222222
SSSSSSSSSSSSSS 000000000 000 RRRRRRRRRRRR RRR TTTTTTTTTTTTTTT 333333333 222222222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSSSSSSSSS 000 000 RRRRRRRRRRRR RRR TTT 333 333 222 222
SSSSSSSSSS 000 000 RRRRRRRRRRRR RRR TTT 333 333 222 222
SSSSSSSSSS 000 000 RRRRRRRRRRRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSS 000 000 RRR RRR TTT 333 333 222 222
SSSSSSSSSSSS 000000000 000 RRR RRR TTT 333333333 222222222222222
SSSSSSSSSSSS 000000000 000 RRR RRR TTT 333333333 222222222222222
SSSSSSSSSSSS 000000000 000 RRR RRR TTT 333333333 222222222222222
```

SSSSSSSS	000000	RRRRRRRR	000000	UU	UU	TTTTTTTTTT	PPPPPPPP	UU	UU	TTTTTTTTTT
SSSSSSSS	000000	RRRRRRRR	000000	UU	UU	TTTTTTTTTT	PPPPPPPP	UU	UU	TTTTTTTTTT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SSSSSS	00	00	RRRRRRRR	00	00	UU	UU	UU	UU	TT
SSSSSS	00	00	RRRRRRRR	00	00	UU	UU	UU	UU	TT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SS	00	00	RR	RR	00	UU	UU	UU	UU	TT
SSSSSSSS	000000	RR	RR	UUUUUUUUUU	UUUUUUUUUU	TT	PP	UUUUUUUUUU	UUUUUUUUUU	TT
SSSSSSSS	000000	RR	RR	UUUUUUUUUU	UUUUUUUUUU	TT	PP	UUUUUUUUUU	UUUUUUUUUU	TT

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLL        IIIII
LLL        IIIII
SSSSSSSS
SSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSS
SSSSSSSS

```

SOF
V04

:

(1) 3
(1) 29

Copyright Notice
Program description

SORSOUTPUT
V04-000

Output text

I 11

16-SEP-1984 01:18:42 VAX/VMS Macro V04-00
5-SEP-1984 03:37:06 [SORT32.SRC]SORSOUTPUT.MAR;1

Page 1
(1)

```
0000 1      .TITLE SORSOUTPUT      Output text
0000 2      .IDENT 'V04-000'      ; File: SORSOUTPUT.MAR Edit: PDG016
0000 3      .SBTTL Copyright Notice
0000 4      :
0000 5      :*****
0000 6      :*
0000 7      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      :*  ALL RIGHTS RESERVED.
0000 10     :*
0000 11     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     :*  TRANSFERRED.
0000 17     :*
0000 18     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     :*  CORPORATION.
0000 21     :*
0000 22     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     :*
0000 25     :*
0000 26     :*****
0000 27     :
```



```

0000 29 .SBTTL Program description
0000 30 :++
0000 31 :
0000 32 FACILITY:
0000 33 :
0000 34 Output text by using $PUTMSG.
0000 35 :
0000 36 ABSTRACT:
0000 37 :
0000 38 NONE
0000 39 :
0000 40 ENVIRONMENT:
0000 41 :
0000 42 Native mode, User mode, AST reentrant
0000 43 :
0000 44 AUTHOR:
0000 45 :
0000 46 Peter D Gilbert, June 1982
0000 47 :
0000 48 MODIFIED BY:
0000 49 :
0000 50 T03-016 PDG016 25-Aug-1982
0000 51 Add a compile-time variable indicating whether to use one
0000 52 $PUTMSG per line, or to use as few $PUTMSG calls as possible.
0000 53 T03-015 Original
0000 54 :--

```

```
0000 56 .DSABL GLOBAL ; Externals must be explicitly declared
0000 57 .EXTRN SYS$FAOL ; System service to format ASCII text
0000 58 .EXTRN SYS$PUTMSG ; System service to output a message
0000 59 $SHRDEF ; Define SHR$_TEXT
0000 60
00000001 0000 61 OPT = 1 ; If non-zero, minimize $PUTMSG calls
0000 62 ; If zero, use one $PUTMSG per line
0000 63
00000000 0000 64 .PSECT SOR$RO_CODE NOVEC,NOWRT,RD,EXE,SHR,LCL,REL,CON,PIC,LONG
0000 65
00000004 0000 66 CTRSTR = 4 ; Offset from AP of control string
00000008 0000 67 PRMLST = 8 ; Offset from AP of parameter list
000000FF 0000 68 K_PUTMSG = 255 ; Maximum size for $PUTMSG
00000400 0000 69 K_BUFSIZ = 1024 ; Buffer size
FFFFFFF8 0000 70 DSC = -8 ; Offset from FP of descriptor
FFFFFFF8 0000 71 LEN = -8 ; Offset from FP of length
FFFFFFFC 0000 72 ADR = -4 ; Offset from FP of address
0A 0D 0000 73 CRLF: .BYTE ^XOD, ^XOA ; Carraige-return, Line-feed
0002 74
003C 0002 75 .ENTRY SOR$SOUTPUT, ^M<R2,R3,R4,R5>
0004 76
0004 77 ; Allocate a descriptor and a buffer on the stack
0004 78
FBF8 CE 9F 0004 79 PUSHAB -K_BUFSIZ-8(SP) ; Push buffer address
00000400 8F DD 0008 80 PUSHL #K_BUFSIZ ; Push buffer length
5E 04 AE D0 000E 81 MOVL 4(SP), SP ; Allocate the buffer
08 AC 9F 0012 82 PUSHAB PRMLST(AP) ; Push address of parameter list
F8 AD 9F 0015 83 PUSHAB DSC(FP) ; Push address of buffer descriptor
F8 AD 9F 0018 84 PUSHAB LEN(FP) ; Push address for returned length
04 AC DD 001B 85 PUSHL CTRSTR(AP) ; Push address of CTRSTR descriptor
00000000'GF 04 FB 001E 86 CALLS #4, G^SYS$FAOL ; Format the output
7A 50 E9 0025 87 BLBC R0, 99$ ; Check status
0028 88
0028 89 ; Form the message vector on the stack
0028 90
F8 AD 9F 0028 91 PUSHAB DSC(FP) ; Push address of descriptor
01 DD 002B 92 PUSHL #^X00000001 ; Push FAO count, Message flags
00091130 8F DD 002D 93 PUSHL #9@16+SHR$_TEXT ; Push message ID
00010003 8F DD 0033 94 PUSHL #^X00010003 ; Push count of longwords, Flags (only message)
0039 95
0039 96 ; Form the argument list on the stack
0039 97
7E 7C 0039 98 CLRQ -(SP) ; Two null parameters
7E D4 003B 99 CLRL -(SP) ; One null parameter
0C AE 9F 003D 100 PUSHAB 12(SP) ; Address of message vector
04 DD 0040 101 PUSHL #4 ; Number of arguments
0042 102
0042 103 ; See whether one $PUTMSG will suffice
0042 104
54 D4 0042 105 10$: CLRL R4 ; Clear the remaining length
00000001 0044 106 .IF NE, OPT
00FF 8F F8 AD B1 0044 107 CMPW LEN(FP), #K_PUTMSG ; Short enough for one $PUTMSG?
43 15 004A 108 BLEQ 90$ ; Branch if short enough
53 FC AD D0 004C 109 MOVL ADR(FP), R3 ; Get address of string
52 000000FF 8F D0 0050 110 MOVL #K_PUTMSG, R2 ; Get maximum length
54 52 7D 0057 111 MOVQ R2, R4 ; Save descriptor of remaining bytes
63 52 A2 AF 02 39 005A 112 MATCHC #2, CRLF, R2, (R3) ; Find a CRLF
```


			F5	13	0060	113		BEQL	20\$; Branch if found
	FC	AD	55	D1	0062	114	30\$:	CMPL	R5, ADR(FP)		; Same address?
			11	13	0066	115		BEQL	40\$; Branch if no CRLF found
50	55	FC	AD	C3	0068	116		SUBL3	ADR(FP), R5, R0		; Compute length to output (plus 2)
54	F8	AD	50	C3	006D	117		SUBL3	R0, LEN(FP), R4		; Compute remaining length
F8	AD	50	02	C3	0072	118		SUBL3	#2, R0, LEN(FP)		; Compute length to output
			16	11	0077	119		BRB	90\$; Go output the text
					0079	120		.ENDC			
			00000001		0079	121		.IF	EQ, OPT		
					0079	122		MOVL	ADR(FP), R5		
					0079	123		.ENDC			
65	F8	AD	83	AF	02	39	0079	MATCHC	#2, CRLF, LEN(FP), (R5)		; Find first CRLF anywhere
			F8	AD	52	C2	0080	SUBL2	R2, LEN(FP)		; Subtract the remaining length
			F8	AD	02	C2	0084	SUBL2	#2, LEN(FP)		; And two bytes for the CRLF
			F8	AD	50	C0	0088	ADDL2	R0, LEN(FP)		; Add two bytes back if CRLF not found
			54	52	7D	008C	128	MOVQ	R2, R4		; Save string descriptor of remainder
00000000	'GF		6E	FA	008F	129	90\$:	CALLG	(SP), G^SYSS\$PUTMSG		; Put out the message
			09	50	E9	0096	130	BLBC	R0, 99\$; Check status
	F8	AD	54	7D	0099	131		MOVQ	R4, DSC(FP)		; Move remainder into the descriptor
			F8	AD	D5	009D	132	TSTL	LEN(FP)		; Is remaining length zero?
			A0	12	00A0	133		BNEQ	10\$; No, go output some more
				04	00A2	134	99\$:	RET			; Return
					00A3	135					
					00A3	136		.END			

SOR\$OUTPUT
Symbol table

Output text

M 11

16-SEP-1984 01:18:42
5-SEP-1984 03:37:06

VAX/VMS Macro V04-00
[SORT32.SRC]SOROUTPUT.MAR;1

Page 5
(2)

ADR = FFFFFFFC
CRLF = 00000000 R 02
CTRSTR = 00000004
DSC = FFFFFFF8
K_BUFSIZ = 00000400
K_PUTMSG = 000000FF
LEN = FFFFFFF8
OPT = 00000001
PRMLST = 00000008
SHR\$ TEXT = 00001130
SOR\$OUTPUT 00000002 RG 02
SY\$FAOL ***** X 00
SY\$PUTMSG ***** X 00

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SOR\$RO_CODE	000000A3 (163.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:00.77
Command processing	119	00:00:00.48	00:00:03.59
Pass 1	130	00:00:01.60	00:00:06.09
Symbol table sort	0	00:00:00.14	00:00:00.51
Pass 2	42	00:00:00.40	00:00:01.33
Symbol table output	3	00:00:00.02	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	327	00:00:02.73	00:00:12.35

The working set limit was 1050 pages.
7520 bytes (15 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 121 non-local and 6 local symbols.
136 source lines were read in Pass 1, producing 13 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2.	4

168 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

SORSOUTPUT
VAX-11 Macro Run Statistics

Output text

N 11

16-SEP-1984 01:18:42 VAX/VMS Macro V04-00
5-SEP-1984 03:37:06 [SORT32.SRC]SORSOUTPUT.MAR;1

Page 6
(2)

MACRO/DISABLE=TRACE/LIS=LISS:SORSOUTPUT/OBJ=OBJ\$:SORSOUTPUT MSRC\$:SORSOUTPUT/UPDATE=(ENH\$:SORSOUTPUT)

0365 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000